

Laboratorio Electrónica Digital

Santiago Ramirez Arenas

Docente: José Alfredo Jaramillo Villegas

Universidad Tecnológica de Pereira

24 de febrero de 2021

Compuertas lógicas

OBJETIVO

En este laboratorio se estudiarán las compuertas lógicas (AND, OR, NOT, XOR, NAND, NOR, XNOR) para crear circuitos lógicos. Se diseñarán estos circuitos usando diferentes enfoques haciendo un uso efectivo de las opciones sintácticas del lenguaje de descripción de hardware SystemVerilog.

Compuertas y circuitos lógicos. En este laboratorio se deberá diseñar y verificar las siguientes Compuertas y circuitos lógicos usando el lenguaje de descripción de hardware SystemVerilog:

PROCEDIMIENTO DEL LABORATORIO

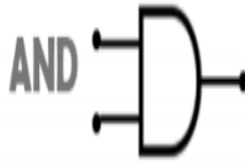
El estudiante deberá realizar de forma individual el diseño de los anteriores circuitos usando el lenguaje de descripción de hardware SystemVerilog en la plataforma en línea de www.edaplayground.com. Luego, deberá diseñar un banco de pruebas usando el mismo lenguaje y realizar la simulación.

Luego de la comprobación del correcto funcionamiento de todas las simulaciones, se deberá hacer un documento de informe de laboratorio en donde se describa los siguientes ítems: • Proceso realizado para el diseño de los circuitos.

- Problemas encontrados durante el diseño con sus respectivas las soluciones.
- Captura de pantalla del diseño.
- Captura de pantalla del banco de pruebas.
- Captura de pantalla de la simulación.
- Interpretación de los resultados de la simulación.
- Conclusiones generales

SOLUCIÓN

Compuerta AND de 2 bit. o Usando una ecuación booleana. o Usando un llamado a un módulo de compuerta lógica.



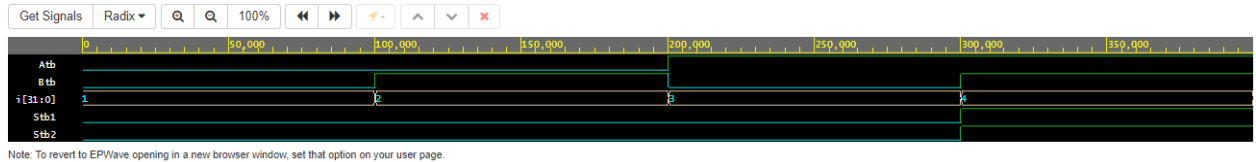
DISEÑO COMPUERTA AND

```
1 module circuito_UNO(  
2     input  logic A,B,  
3     output logic S1,S2);  
4  
5     assign S1 = (A & B) ;  
6     and comp1 (S2,A,B);  
7 endmodule
```

TESTBENCH COMPUERTA AND

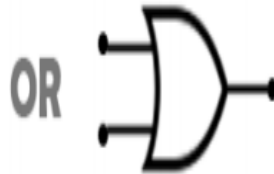
```
1 module test_circuito_UNO ();  
2     logic Atb=0;  
3     logic Btb=0;  
4     logic Stb1;  
5     logic Stb2;  
6     integer i=1;  
7  
8     circuito_UNO circuito (  
9         .A(Atb),  
10        .B(Btb),  
11        .S1(Stb1),  
12        .S2(Stb2));  
13  
14     initial begin  
15         $dumpfile("dump.vcd");  
16         $dumpvars(1,test_circuito_UNO);  
17         for(i=1;i<5;i=i+1)  
18             begin  
19                 #100  
20                 if(i==2) begin  
21                     Atb=1;  
22                 end  
23                 Btb=~Btb;  
24             end  
25             $finish;  
26     end  
27  
28 endmodule
```

SIMULACIÓN COMPUERTA AND



Compuerta OR de 2 bit.

- o Usando una ecuación booleana.
- o Usando un llamado a un módulo de compuerta lógica.



DISEÑO COMPUERTA OR

```
1 module circuito_DOS (
2     input A,B,
3     output S1,S2);
4
5
6     assign S1 = (A | B) ;
7     or comp1 (S2,A,B);
8
9 endmodule
```

TESTBENCH COMPUERTA OR

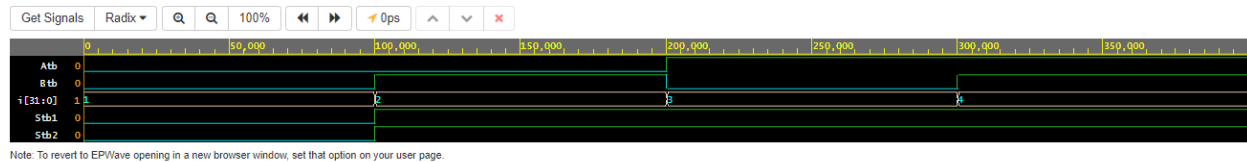
```
1 module test_circuito_DOS ();
2     logic Atb=0;
3     logic Btb=0;
4     logic Stb1;
5     logic Stb2;
6     integer i=1;
7
8     circuito_DOS circuito (
9         .A(Atb),
10        .B(Btb),
11        .S1(Stb1),
12        .S2(Stb2));
13
14    initial begin
15        $dumpfile("dump.vcd");
16        $dumpvars(1,test_circuito_DOS);
17        for(i=1;i<5;i=i+1)
18            begin
19                #100
20                if(i==2) begin
```

```

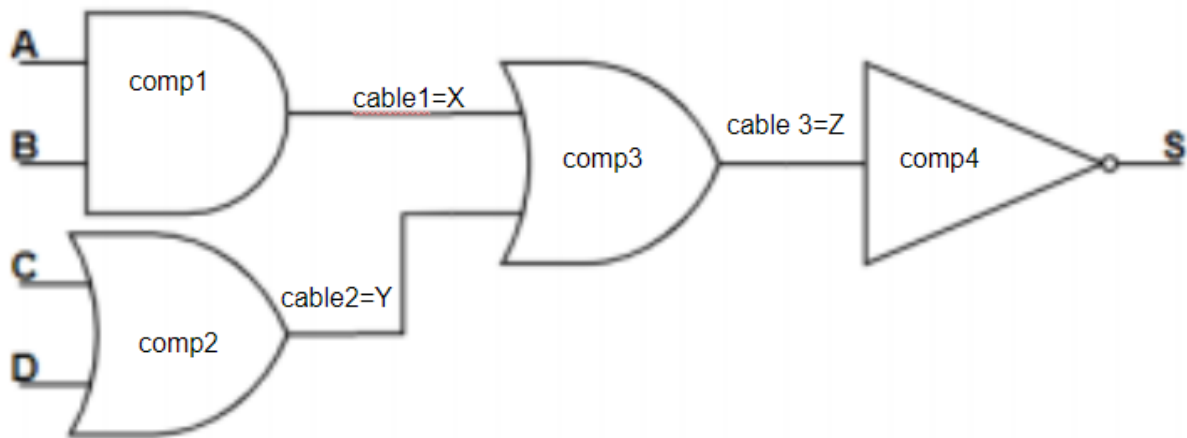
21     Atb=1;
22     end
23     Btb=~Btb;
24     end
25     $finish;
26 end
27
28 endmodule

```

SIMULACIÓN COMPUERTA OR



- Circuito lógico entradas 4 a 1 salida
 - o Usando ecuaciones booleanas.
 - o Usando llamados a módulos de compuertas lógicas.



DISEÑO CIRCUITO TRES

```

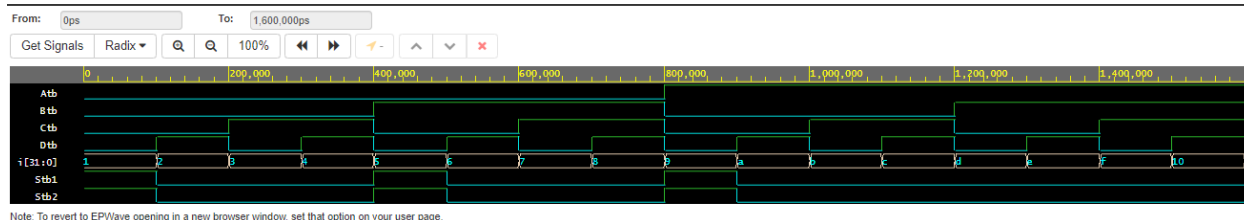
1
2 module circuito_TRES(
3     input logic A,B,C,D,
4     output logic S1,S2);
5     wire X,Y,Z;
6
7     assign S1 = ! ((A & B ) | (C | D ));
8
9     and comp1 (X,A,B);
10    or comp2 (Y,C,D);
11    or comp3 (Z,X,Y);
12    not comp4 (S2,Z);
13
14 endmodule

```

TESTBENCH CIRCUITO TRES

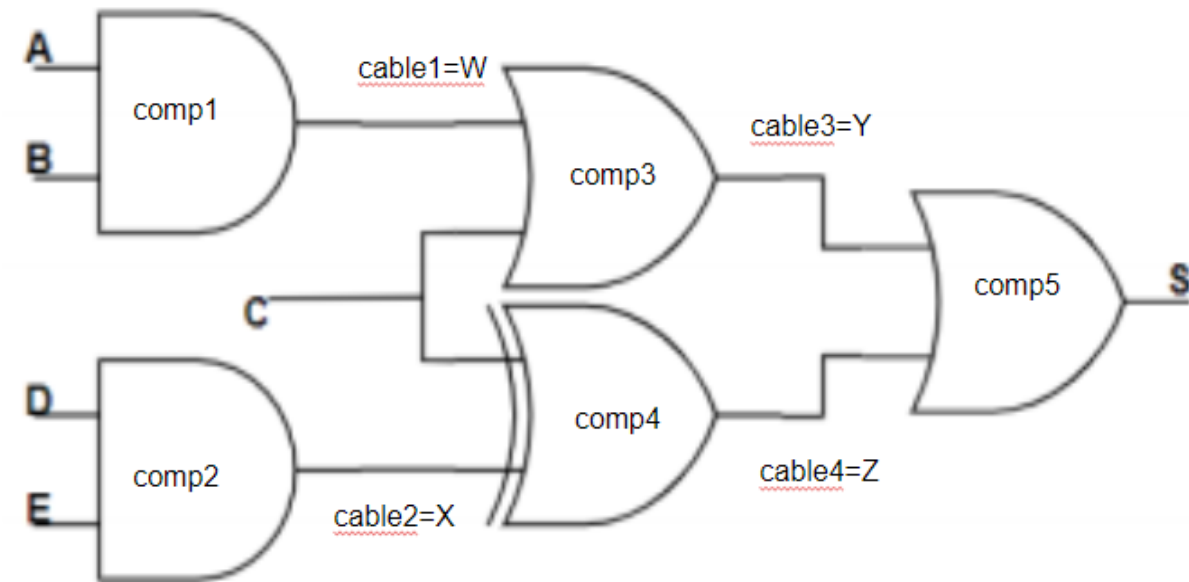
```
1 module test_circuito_TRES();
2
3
4   logic Atb=0;
5   logic Btb=0;
6   logic Ctb=0;
7   logic Dtb=0;
8   logic Stb1;
9   logic Stb2;
10  integer i=1;
11
12  circuito_TRES circuito (
13    .A(Atb),
14    .B(Btb),
15    .C(Ctb),
16    .D(Dtb),
17    .S1(Stb1),
18    .S2(Stb2));
19
20  initial begin
21    $dumpfile("dump.vcd");
22    $dumpvars(1,test_circuito_TRES);
23    for(i=1;i<17;i=i+1)
24      begin
25        #100
26        if(i==8) begin
27          Atb=1;
28        end
29        if(i%4==0) begin
30          Btb=~Btb;
31        end
32        if(i%2==0) begin
33          Ctb=~Ctb;
34        end
35        Dtb=~Dtb;
36      end
37    $finish;
38  end
39
40 endmodule
```

SIMULACIÓN CIRCUITO TRES



- Circuito lógico entradas 5 a 1 salida

- o Usando ecuaciones booleanas.
- o Usando llamados a módulos de compuertas lógicas.



DISEÑO CIRCUITO CUATRO

```

1
2 module circuito_CUATRO(
3   input A,B,C,D,E,
4   output S1,S2);
5   wire W,X,Y,Z;
6
7   assign S1 = ( (A & B) | C ) | ((D & E) ^ C );
8
9   and comp1 (W,A,B);
10  and comp2 (X,D,E);
11  or comp3 (Y,W,C);
12  xor comp4 (Z,X,C);
13  or comp5 (S2,Y,Z);
14
15 endmodule

```

TESTBENCH CIRCUITO CUATRO

```

1 module test_circuito_CUATRO;
2
3   logic Atb=0;
4   logic Btb=0;
5   logic Ctb=0;
6   logic Dtb=0;
7   logic Etb=0;
8   logic Stb1;
9   logic Stb2;

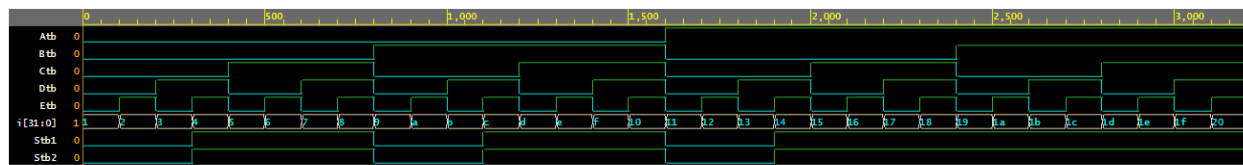
```

```

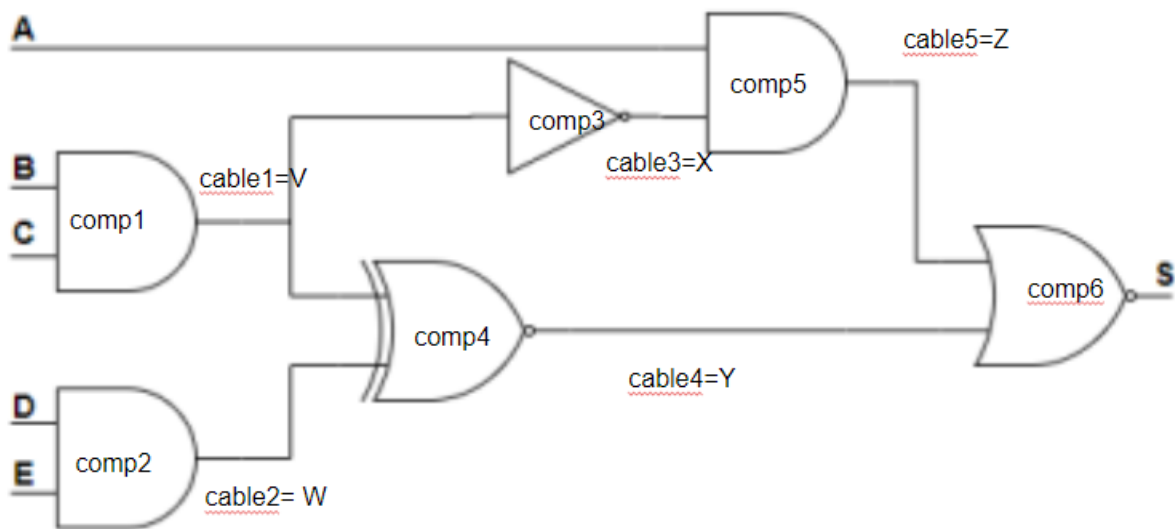
10  integer i=1;
11
12  circuito_CUATRO circuito (
13      .A(Atb),
14      .B(Btb),
15      .C(Ctb),
16      .D(Dtb),
17      .E(Etb),
18      .S1(Stb1),
19      .S2(Stb2));
20
21  initial begin
22      $dumpfile("dump.vcd");
23      $dumpvars(1,test_circuito_CUATRO);
24      for(i=1;i<33;i=i+1)
25          begin
26              #100
27              if(i==16) begin
28                  Atb=1;
29              end
30              if(i%8==0) begin
31                  Btb=~Btb;
32              end
33              if(i%4==0) begin
34                  Ctb=~Ctb;
35              end
36              if(i%2==0) begin
37                  Dtb=~Dtb;
38              end
39              end
40              Etb=~Etb;
41          end
42      $finish;
43  end
44
45  endmodule

```

SIMULACIÓN CIRCUITO CUATRO



- Circuito lógico entradas 5 a 1 salida
- o Usando ecuaciones booleanas.
- o Usando llamados a módulos de compuertas lógicas.



DISEÑO CIRCUITO CINCO

```

1
2 module circuito_CINCO(
3   input A,B,C,D,E,
4   output S1,S2);
5   wire V,W,X,Y,Z;
6
7   assign S1 = ((B & C) ^ (D & E)) ~| (~(B & C) & A);
8
9   and comp1 (V,B,C);
10  and comp2 (W,D,E);
11  not comp3 (X,V);
12  xnor comp4 (Y,V,W);
13  and comp5 (Z,A,X);
14  nor comp6 (S2,Y,Z);
15
16 endmodule

```

TESTBENCH CIRCUITO CINCO

```

1
2 module test_circuito_CINCO;
3
4   logic Atb=0;
5   logic Btb=0;
6   logic Ctb=0;
7   logic Dtb=0;
8   logic Etb=0;
9   logic Stb1;
10  logic Stb2;
11  integer i=1;
12
13  circuito_CINCO circuito (
14    .A(Atb),

```



```

15     .B(Btb),
16     .C(Ctb),
17     .D(Dtb),
18     .E(Etb),
19     .S1(Stb1),
20     .S2(Stb2));
21
22 initial begin
23     $dumpfile("dump.vcd");
24     $dumpvars(1,test_circuito_CINCO);
25     for(i=1;i<33;i=i+1)
26         begin
27             #100
28             if(i==16) begin
29                 Atb=1;
30             end
31             if(i%8==0) begin
32                 Btb=~Btb;
33             end
34             if(i%4==0) begin
35                 Ctb=~Ctb;
36             end
37             if(i%2==0) begin
38                 Dtb=~Dtb;
39             end
40             Etb=~Etb;
41         end
42     $finish;
43 end
44
45 endmodule

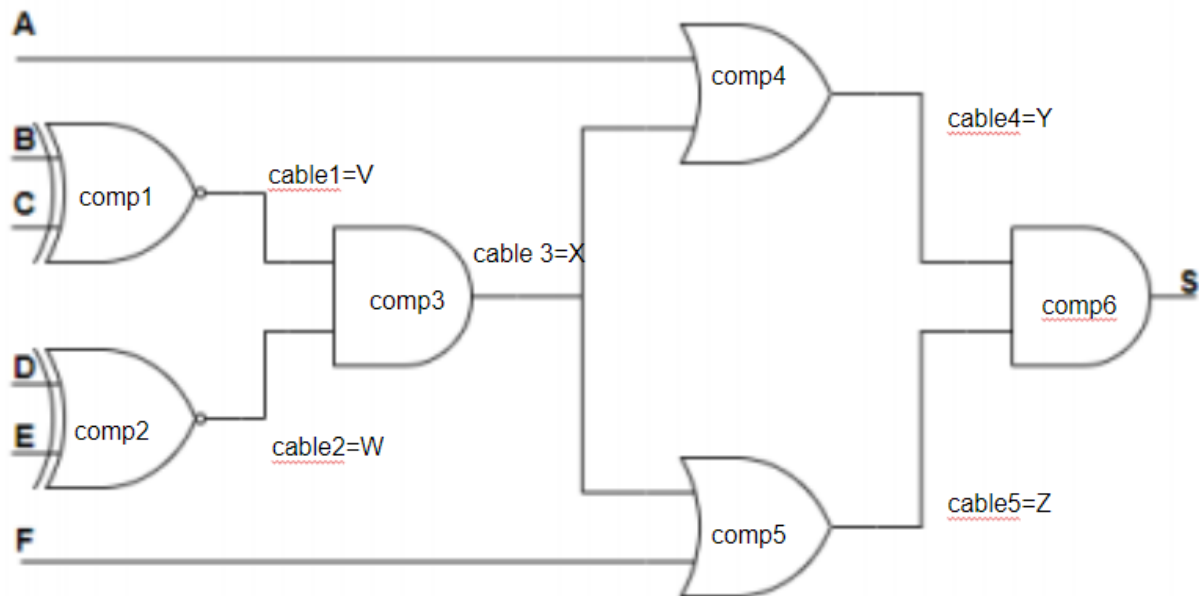
```

SIMULACIÓN CIRCUITO CINCO



Note: To revert to EPWAVE opening in a new browser window, set that option on your user page.

- Circuito lógico entradas 6 a 1 salida
 - o Usando ecuaciones booleanas.
 - o Usando llamados a módulos de compuertas lógicas.



DISEÑO CIRCUITO SEIS

```

1 module    circuito_SEIS(
2   input  A,B,C,D,E,F,
3   output S1,S2);
4   wire  V,W,X,Y,Z;
5
6   assign  S1 = ( A | ((B  ~^  C) & (D  ~^  E)) ) & ( F | ((B  ~^  C) &
7     (D  ~^  E)) ) ;
8
9   xnor  comp1      (V,B,C);
10  xnor  comp2      (W,D,E);
11  and   comp3      (X,V,W);
12  or    comp4      (Y,A,X);
13  or    comp5      (Z,F,X);
14  and   comp6      (S2,Y,Z);
15 endmodule

```

TESTBENCH CIRCUITO SEIS

```

1 module    test_circuito_SEIS;
2
3   logic  Atb=0;
4   logic  Btb=0;
5   logic  Ctb=0;
6   logic  Dtb=0;
7   logic  Etb=0;
8   logic  Ftb=0;
9   logic  Stb1;
10  logic  Stb2;
11  integer i=1;
12
13  circuito_SEIS  circuito (

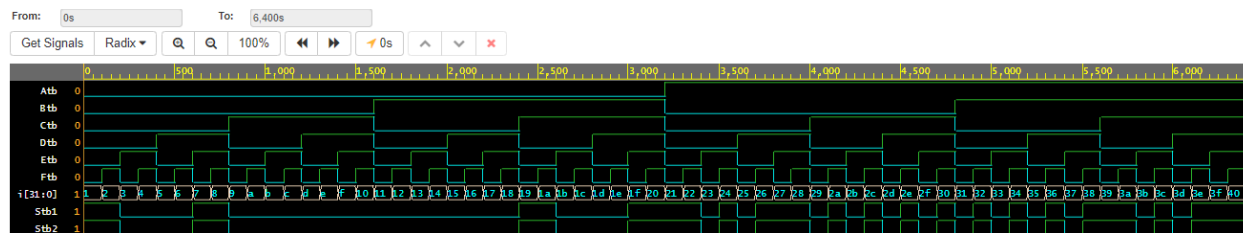
```

```

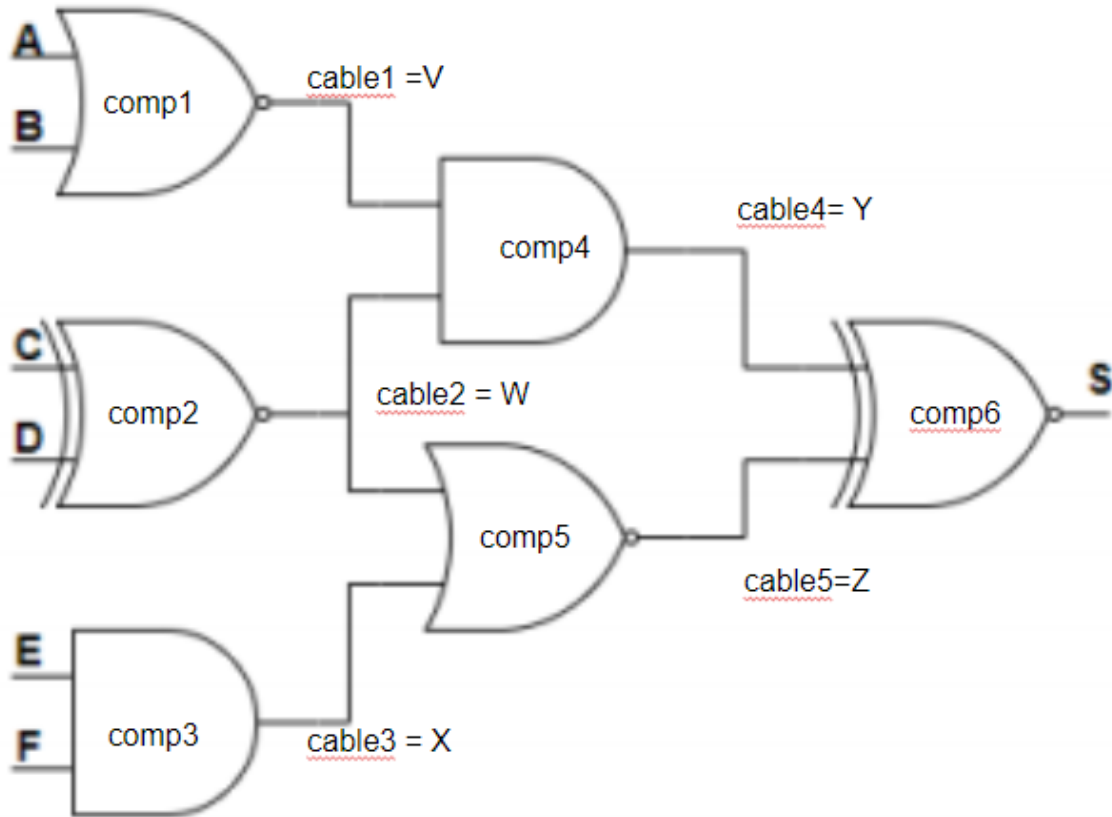
14     .A(Atb),
15     .B(Btb),
16     .C(Ctb),
17     .D(Dtb),
18     .E(Etb),
19     .F(Ftb),
20     .S1(Stb1),
21     .S2(Stb2));
22
23 initial begin
24     $dumpfile("dump.vcd");
25     $dumpvars(1,test_circuito_SEIS);
26     for(i=1;i<65;i=i+1)
27         begin
28             #100
29             if(i==32) begin
30                 Atb=1;
31             end
32             if(i%16==0) begin
33                 Btb=~Btb;
34             end
35             if(i%8==0) begin
36                 Ctb=~Ctb;
37             end
38             if(i%4==0) begin
39                 Dtb=~Dtb;
40             end
41             if(i%2==0) begin
42                 Etb=~Etb;
43             end
44             Ftb=~Ftb;
45         end
46     $finish;
47 end
48
49 endmodule

```

SIMULACIÓN CRICUITO SEIS



- Circuito lógico entradas 6 a 1 salida
- o Usando ecuaciones booleanas.
 - o Usando llamados a módulos de compuertas lógicas.



DISEÑO CIRCUITO SIETE

```

1 module    circuito_SIETE(
2   input  A,B,C,D,E,F,
3   output S1,S2);
4   wire  V,W,X,Y,Z;
5
6   assign  S1 = ((A ~| B) & (C ~^ D) ) ~^ ((C ~^ D) ~| (E & F));
7
8   nor     comp1    (V,A,B);
9   xnor    comp2    (W,C,D);
10  and     comp3    (X,E,F);
11  and     comp4    (Y,V,W);
12  nor     comp5    (Z,W,X);
13  xnor    comp6    (S2,Y,Z);
14
15 endmodule

```

TESTBENCH CIRCUITO SIETE

```

1 module    test_circuito_SIETE;
2
3   logic  Atb=0;
4   logic  Btb=0;
5   logic  Ctb=0;
6   logic  Dtb=0;
7   logic  Etb=0;
8   logic  Ftb=0;

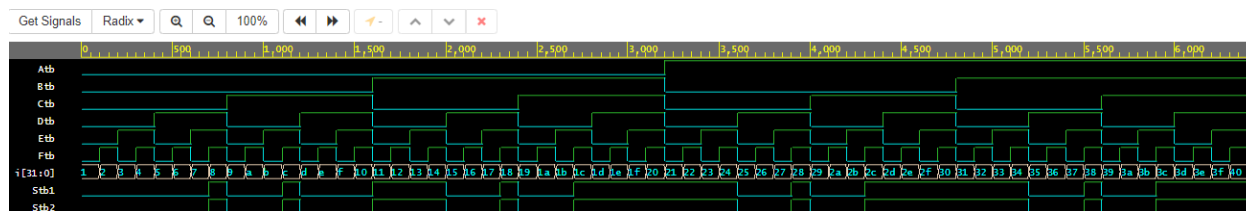
```

```

9  logic Stb1;
10 logic Stb2;
11 integer i=1;
12
13 circuito_SIETE circuito (
14     .A(Atb),
15     .B(Btb),
16     .C(Ctb),
17     .D(Dtb),
18     .E(Etb),
19     .F(Ftb),
20     .S1(Stb1),
21     .S2(Stb2));
22
23 initial begin
24     $dumpfile("dump.vcd");
25     $dumpvars(1,test_circuito_SIETE);
26     for(i=1;i<65;i=i+1)
27         begin
28             #100
29             if(i==32) begin
30                 Atb=1;
31             end
32             if(i%16==0) begin
33                 Btb=~Btb;
34             end
35             if(i%8==0) begin
36                 Ctb=~Ctb;
37             end
38             if(i%4==0) begin
39                 Dtb=~Dtb;
40             end
41             if(i%2==0) begin
42                 Etb=~Etb;
43             end
44             Ftb=~Ftb;
45         end
46     $finish;
47 end
48
49 endmodule

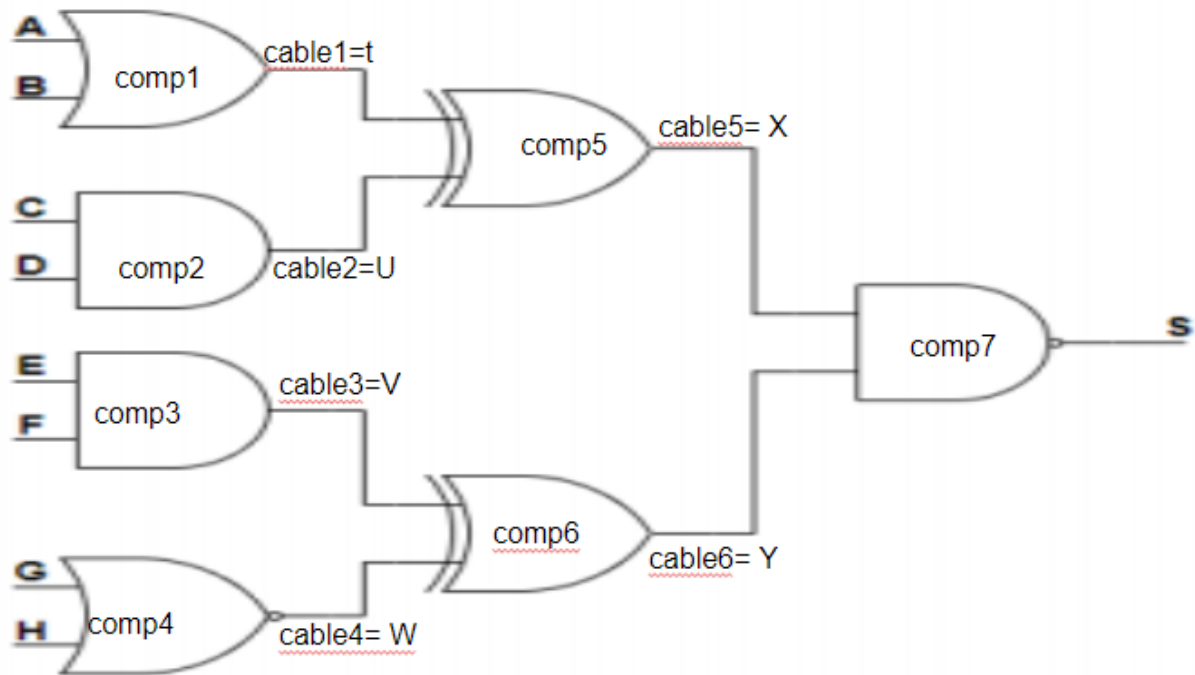
```

SIMULACIÓN CIRCUITO SIETE



- Circuito lógico entradas 8 a 1 salida
- o Usando ecuaciones booleanas.

o Usando llamados a módulos de compuertas lógicas.



DISEÑO CIRCUITO OCHO

```

1 module    circuito_OCHO(
2   input  A,B,C,D,E,F,G,H,
3   output S1,S2);
4   wire  T,U,V,W,X,Y;
5
6   assign  S1 = ((A | B) ^ (C & D)) ~& ((E & F)^(G ~|H));
7
8   or  comp1 (T,A,B);
9   and comp2 (U,C,D);
10  and comp3 (V,E,F);
11  nor comp4 (W,G,H);
12  xor comp5 (X,T,U);
13  xor comp6 (Y,V,W);
14  nand comp7 (S2,X,Y);
15
16 endmodule

```

TESTBENCH CIRCUITO OCHO

```

1 module    test_circuito_OCHO;
2
3   logic  Atb=0;
4   logic  Btb=0;
5   logic  Ctb=0;
6   logic  Dtb=0;
7   logic  Etb=0;

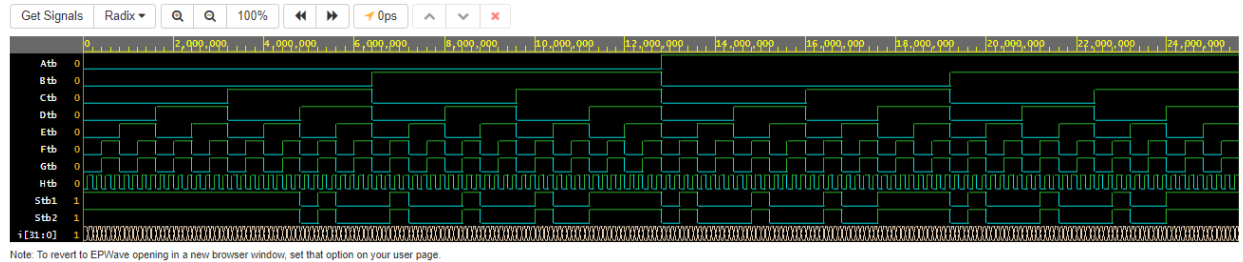
```

```

8  logic Ftb=0;
9  logic Gtb=0;
10 logic Htb=0;
11 logic Stb1;
12 logic Stb2;
13 integer i=1;
14
15 circuito_OCHO circuito (
16     .A(Atb),
17     .B(Btb),
18     .C(Ctb),
19     .D(Dtb),
20     .E(Etb),
21     .F(Ftb),
22     .G(Gtb),
23     .H(Htb),
24     .S1(Stb1),
25     .S2(Stb2));
26
27 initial begin
28     $dumpfile("dump.vcd");
29     $dumpvars(1,test_circuito_OCHO);
30     for(i=1;i<257;i=i+1)
31         begin
32             #100
33             if(i==128) begin
34                 Atb=1;
35             end
36             if(i%64==0) begin
37                 Btb=~Btb;
38             end
39             if(i%32==0) begin
40                 Ctb=~Ctb;
41             end
42             if(i%16==0) begin
43                 Dtb=~Dtb;
44             end
45             if(i%8==0) begin
46                 Etb=~Etb;
47             end
48             if(i%4==0) begin
49                 Ftb=~Ftb;
50             end
51             if(i%4==0) begin
52                 Gtb=~Gtb;
53             end
54             Htb=~Htb;
55         end
56     $finish;
57 end
58
59 endmodule

```

SIMULACIÓN CIRCUITO OCHO



Desarrollo

- Proceso realizado para el diseño de los circuitos.
 - Inicialmente se debe reconocer cada tipo de compuerta que hace parte del circuito lógico.
 - Después se deben nombrar las conexiones que tiene cada circuito, visto de otra forma los cables, o salidas de cada compuerta lógica, esto es importante para escribir de manera correcta tanto el llamado a las compuertas lógicas como la elaboración de las fórmulas booleanas que describen cada circuito.
 - Es importante definir cada tipo de dato, ya sean entradas, números enteros o las conexiones, por ejemplo:
 - reg: Representan variables con capacidad de almacenar información.
 - wire: Representan conexiones estructurales entre componentes. No tienen capacidad de almacenamiento.
 - integer: Registro de 32 bits.
 - Ya finalizado el diseño lo siguiente es el "testbench", nuevamente declaramos tanto salidas como entradas, teniendo presente el número de entradas al momento de escribir las posibles combinaciones entre N número de entradas, algo que facilita bastante el trabajo es el uso de ciclos for y condicionales if esto cuando tenemos una gran número de entradas dado que se vuelve tedioso escribir cada combinación manualmente.
 - Finalmente visualizamos las simulaciones y comparamos los valores de ambas salidas, Ecuaciones Booleanas y ,Llamado a módulos de compuertas lógicas: (S1 y S2 respectivamente)
 - Interpretación de los resultados de la simulación.
 - De acuerdo a la simulación, pude observar que algunos circuitos tenían los mismos patrones con respecto al número de entradas
- Conclusiones generales.
 - El reconocimiento del tipo de datos que se van a usar y la declaración de cada uno de ellos es fundamental al momento de hacer un diseño de un circuito lógico.
 - El uso de ciclos y condicionales facilitan en gran medida el tiempo invertido en la creación del testbench cuando se trabaja con un gran número de variables.

- El diseño por el método de compuertas lógicas es más fácil en comparación con las Ecuaciones Booleanas al momento de crear un diseño de un circuito lógico.
- Problemas encontrados durante el diseño con sus respectivas las soluciones.
- Uno de los problemas con los que me encontré fue el de implementar un ciclo for para crear las posibles combinaciones en el testbench, dado que es muy tedioso hacerlo manualmente y más aún cuando se trabajan con un número moderado de entradas. La solución fue dada por el docente, adicionalmente consulté un poco la sintaxis en:

<https://es.wikibooks.org/wiki/Programacion>